ФПМИ БГУ

Вычислительные методы алгебры
Лаборатоная работа 10

Нахождение собственных значений и собственных векторов степенным методом

Подготовил:
Ткачук Павел
2 курс 1 группа

Преподаватель:
Будник Анатолий Михайлович

21 декабря 2016 г.

1 Постановка задачи

Входные данные:

$$A_{\text{исх.}} = \begin{bmatrix} 0.6444 & 0.0000 & -0.1683 & 0.1184 & 0.1973 \\ -0.0395 & 0.4208 & 0.0000 & -0.0802 & 0.0263 \\ 0.0132 & -0.1184 & 0.7627 & 0.0145 & 0.0460 \\ 0.0395 & 0.0000 & -0.0960 & 0.7627 & 0.0000 \\ 0.0263 & -0.0395 & 0.1907 & -0.0158 & 0.5523 \end{bmatrix}$$

Для того чтобы все собственные значения были действительными числами, домножим слева исходную матрицу на ей транспонированную:

$$A = A_{\text{исх.}} A^T_{\text{исх.}} = \begin{bmatrix} 0.4965221 & -0.02976049 & -0.10906373 & 0.13191428 & 0.09195098 \\ -0.02976049 & 0.18575662 & -0.05029722 & -0.06272879 & -0.0018678 \\ -0.10906373 & -0.05029722 & 0.59823034 & -0.06163865 & 0.17564755 \\ 0.13191428 & -0.06272879 & -0.06163865 & 0.59248754 & -0.02931901 \\ 0.09195098 & -0.0018678 & 0.17564755 & -0.02931901 & 0.34390336 \end{bmatrix}$$

Задача:

1. Для данной матрицы A вычислить максимальное по модулю собственное значение λ_{max}, собственный вектор $\vec{v}$ соответствующий максимальному значению, а также количество итераций за которое было получено решение. $\varepsilon = 10^{-5}$
2. Найти невязку $\vec{r} = A\vec{v} - \lambda_{max}\vec{v}$

2 Алгоритм

1. Полагаем $k = 0$, $y^{(0)} = (1, 1, \ldots, 1)$, $\lambda^{(0)} = 1$ и $\varepsilon = 10^{-5}$
2. Вычисляем следующее приближение

$$y^{(k+1)} = Ay^{(k)},$$
$$\lambda^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \frac{y_i^{(k+1)}}{y_i^{(k)}}$$

3. Если выполнено условие $\|\lambda^{(k+1)} - \lambda^{(k)}\| < \epsilon$, завершаем процесс и в качестве приближенного собственного значения берем $\lambda^* \cong \lambda^{(k+1)}$, а в качестве собственного вектора $\vec{v} \cong y^{(k+1)}$. Иначе нормируем $y^{(k+1)}$ полагаем $k = k + 1$, и переходим к пункту 2 алгоритма.

3 Результаты и вывод

3.1 Входные данные

0.6444 0.0000 -0.1683 0.1184 0.1973
-0.0395 0.4208 0.0000 -0.0802 0.0263
0.0132 -0.1184 0.7627 0.0145 0.0460
0.0395 0.0000 -0.0960 0.7627 0.0000
0.0263 -0.0395 0.1907 -0.0158 0.5523

3.2 Выходные данные

```
Ищем собственные вектор матрицы
Исходная матрица:
[[ 0.4965221  -0.02976049 -0.10906373  0.13191428  0.09195098]
 [-0.02976049  0.18575662 -0.05029722 -0.06272879 -0.0018678 ]
 [-0.10906373 -0.05029722  0.59823034 -0.06163865  0.17564755]
```

```
 [ 0.13191428 -0.06272879 -0.06163865  0.59248754 -0.02931901]
 [ 0.09195098 -0.0018678   0.17564755 -0.02931901  0.34390336]]
Собственное значение:  0.780801286504
Собственный вектор:  [ 0.45192578 -0.02699165 -0.6529953   0.57128303 -0.20559032]
Количество итераций:  49
Невязка:  [  4.45401707e-06   3.66055326e-06  -3.76876208e-05   5.62171434e-06
  -1.90591697e-05]
Норма невязки:  4.29936076537e-05
```

3.3 Вывод

Данный метод относится к степенным. С его помощью можно получить максимальное собственное значение с любой точностью, однако количество шагов, требуемое для получения достаточно точного результата (в данном случае с точностью $\varepsilon = 10^{-5}$), значительно меньше, чем в точных методах. Можно сделать вывод, что во многих задачах, где требуется получить собственные значения с определенным свойством, выгоднее использовать степенные методы так как можно получить решение с такой же погрешностью как и в точных методах (в точных методах появляется погрешность вычислений), однако за меньшее количество операций.

4 Листинг кода

```
import numpy as np
import numpy.linalg as linalg

def eigvals(matr_A, eps):
    A = matr_A.copy()
    n = len(A)
    y = np.array([1 for i in range(n)])
    converge = False
    count = 0
    prev = 1
    while not converge:
        count +=1
        y_new = np.dot(A, y)
        next = sum(y_new[i] / y[i] for i in range(n)) / n
        converge = abs(next - prev) <= eps
        prev = next
        y = y_new / linalg.norm(y_new)
    return y, prev, count

file = open("matrix", "r")  # Чтение файла
A, b = [], []
for line in file:
    A.append([float(el) for el in line.split()[:-1]])
    b.append(float(line.split().pop()))
A = np.array(A)
A = np.dot(A, A.transpose())
b = np.array(b)
print("Ищем собственные вектор матрицы")
print("Исходная матрица:")
print(A)
ans = eigvals(A, 0.00001)
print("Собственное значение: ", ans[1])
print("Собственный вектор: ", ans[0])
print("Количество итераций: ", ans[2])
print("Невязка: ", np.dot(A, ans[0]) - ans[0] * ans[1])
print("Норма невязки: ", linalg.norm(np.dot(A, ans[0]) - ans[0] * ans[1]))
```